

1

## EXCLUSION CONTROL

2

### FIELD OF INVENTION

3

4

5

6

7

8

The present invention relates to an exclusion controller, an exclusion control method, a program, and a recording medium. In particular, the present invention relates to an exclusion controller, an exclusion control method, a program, and a recording medium, which give priority for acquiring a contended resource to a specific information processing unit.

9

### BACKGROUND OF THE INVENTION

10

11

12

13

14

15

16

17

In recent years, multiprocessors and multithreads are used as technologies for causing a computer to execute a plurality of tasks in parallel and for causing a plurality of processors to cooperatively execute a task at a high speed. In these technologies, a plurality of information processing units, such as processors or threads, share a contended resource such as a memory or an input/output device.

Heretofore, a method for enabling high-speed acquisition and

1 release of a contended resource by changing an algorithm used for  
2 arbitration of the contention depending on whether the degree of  
3 contention is relatively low or high when the arbitration for the  
4 contended resource is performed, has been proposed (refer to  
5 nonpatent literature 1). According to the method disclosed in  
6 nonpatent literature 1, a contended resource, in particular, can be  
7 acquired and released at a high speed compared to other methods.

8 (Nonpatent Literature 1)

9 Tamiya Onodera and Kiyokuni Kawachiya: "A Study of Locking  
10 Objects with Bimodal Fields," in Proceedings of OOPSLA '99.

11 However, there are cases where only a specific information  
12 processing unit among a plurality of information processing units  
13 frequently acquires and releases a contended resource. According to  
14 known technologies, even in such cases, the specific information  
15 processing unit needs to execute a process while operating to the  
16 exclusion of the other information processing units in the  
17 acquisition of the contended resource. This has made it difficult  
18 to improve a processing speed.

19 SUMMARY OF THE INVENTION

1           Accordingly, the present invention provides an exclusion  
2 controller, an exclusion control method, a program, and a recording  
3 medium, which solve the above-described problem. Specifically, a  
4 first aspect of the present invention provides an exclusion  
5 controller which allows an information processing unit to acquire a  
6 contended resource to the exclusion of other information processing  
7 units. In an example embodiment, the exclusion controller includes:  
8 a plurality of non-prioritized information processing units mutually  
9 exclusively acquiring a non-prioritized exclusion right, which  
10 indicates a candidate for acquiring the contended resource, by a  
11 first process; and a prioritized information processing unit  
12 acquiring the contended resource by a second process, which requires  
13 a shorter processing time than the first process, to the exclusion  
14 of the non-prioritized information processing unit having acquired  
15 the non-prioritized exclusion right.

16           A second aspect of the present invention provides an exclusion  
17 controller which allows any one of a plurality of threads capable of  
18 acquiring an identical contended resource to acquire the contended  
19 resource to the exclusion of the other threads different from the  
20 relevant thread. The exclusion controller includes: an execution  
21 state acquisition/notification unit for acquiring execution location  
22 information indicating an execution location of a program in a first

1 thread of the threads and for notifying a second thread different  
2 from the first thread of the execution location information; and an  
3 execution state setting unit for allowing the second thread to  
4 execute a process for setting, in the first thread, execution  
5 location information indicating that the contended resource is not  
6 being acquired if the acquired execution location information  
7 indicates that the contended resource is being acquired. Further,  
8 the second aspect of the present invention provides an exclusion  
9 control method, a program for realizing the method, and a recording  
10 medium having the program recorded thereon.

#### 11 BRIEF DESCRIPTION OF THE DRAWINGS

12 For a more complete understanding of the present invention and  
13 the advantages thereof, reference is now made to the following  
14 description taken in conjunction with the accompanying drawings, in  
15 which:

16 Fig. 1 is a function block diagram of an exclusion controller  
17 10 in a first embodiment;

18 Fig. 2 is a view showing details of a flat lock storage unit  
19 100 in the first embodiment;

1           Fig. 3 is an operation flowchart where a prioritized  
2 information processing unit 150 or a non-prioritized information  
3 processing unit 160 tries to acquire a contended resource in the  
4 first embodiment;

5           Fig. 4 is an operation flowchart showing details of S100A and  
6 S100B in Fig. 3;

7           Fig. 5 is an operation flowchart showing details of S170 in  
8 Fig. 3;

9           Fig. 6 is an operation flowchart where the prioritized  
10 information processing unit 150 or the non-prioritized information  
11 processing unit 160 releases the contended resource;

12           Fig. 7 is a function block diagram of an exclusion controller  
13 10 in a modified example of the first embodiment;

14           Fig. 8 is a view showing details of a flat lock storage unit  
15 100 in the modified example of the first embodiment;

16           Fig. 9 is an operation flowchart where a prioritized  
17 information processing unit 150 or a non-prioritized information  
18 processing unit 160 tried to acquire a contended resource in the  
19 modified example of the first embodiment;

20           Fig. 10 is a function block diagram of an exclusion controller  
21 20 in a second embodiment;

22           Fig. 11 is a view showing details of a resource information

1 storage area 300 in the second embodiment;

2 Fig. 12 is an operation flowchart where a prioritized  
3 information processing unit 340 or a non-prioritized information  
4 processing unit 350 tries to acquire a contended resource in the  
5 second embodiment;

6 Fig. 13 is an operation flowchart showing details of S880 in  
7 Fig. 12;

8 Fig. 14 is an operation flowchart where the prioritized  
9 information processing unit 340 or the non-prioritized information  
10 processing unit 350 releases the contended resource in the second  
11 embodiment;

12 Fig. 15 is an operation flowchart showing details of S880 in a  
13 modified example of the second embodiment; and

14 Fig. 16 is a view showing an example of the hardware  
15 configuration of the exclusion controller 10.

## 16 DETAILED DESCRIPTION OF THE INVENTION

17 The present invention provides an exclusion controller, an  
18 exclusion control method, a program, and a recording medium. In an  
19 embodiment, the present invention provides an exclusion controller

1    which allows an information processing unit to acquire a contended  
2    resource to the exclusion of other information processing units.  
3    The exclusion controller includes: a plurality of non-prioritized  
4    information processing units mutually exclusively acquiring a  
5    non-prioritized exclusion right, which indicates a candidate for  
6    acquiring the contended resource, by a first process; and a  
7    prioritized information processing unit acquiring the contended  
8    resource by a second process, which requires a shorter processing  
9    time than the first process, to the exclusion of the non-prioritized  
10   information processing unit having acquired the non-prioritized  
11   exclusion right. Thus, the present invention provides an exclusion  
12   control method, a program for realizing the method, and a recording  
13   medium having the program recorded thereon.

14           Moreover, the present invention provides an exclusion  
15   controller which allows any one of a plurality of threads capable of  
16   acquiring an identical contended resource to acquire the contended  
17   resource to the exclusion of the other threads different from the  
18   relevant thread. The exclusion controller includes: an execution  
19   state acquisition/notification unit for acquiring execution location  
20   information indicating an execution location of a program in a first  
21   thread of the threads and for notifying a second thread different  
22   from the first thread of the execution location information; and an

1 execution state setting unit for allowing the second thread to  
2 execute a process for setting, in the first thread, execution  
3 location information indicating that the contended resource is not  
4 being acquired if the acquired execution location information  
5 indicates that the contended resource is being acquired. Further,  
6 the second aspect of the present invention provides an exclusion  
7 control method, a program for realizing the method, and a recording  
8 medium having the program recorded thereon.

9 Hereinafter, the present invention will be described through  
10 particular example embodiments thereof. However, the embodiments  
11 below are not intended to limit the invention according to the  
12 appended claims. Moreover, all combinations of features described  
13 in the embodiments are not always necessary for solving means of the  
14 invention.

15 (First Embodiment)

16 Figure 1 shows a function block diagram of an exclusion  
17 controller 10 in a first embodiment. The exclusion controller 10  
18 has an object to arbitrate the acquisition of a contended resource  
19 50 by allowing any one of a plurality of information processing  
20 units operating parallel to one another, for example, a prioritized  
21 information processing unit 150 and non-prioritized information



1 processing units 160-1 to 160-N, to acquire the contended resource  
2 50 to the exclusion of the other information processing units.

3 The exclusion controller 10 comprises the contended resource  
4 50 to be acquired by any one of the plurality of information  
5 processing units, a flat lock storage unit 100 used when the degree  
6 of contention for the contended resource 50 is relatively low, a  
7 prioritized information processing unit setting unit 140 for setting  
8 one of the plurality of information processing units as the  
9 prioritized information processing unit 150, the prioritized  
10 information processing unit 150 given priority in acquisition of the  
11 contended resource 50, non-prioritized information processing units  
12 160-1 to 160-N, an acquisition check unit 170 for checking the  
13 degree of contention for the contended resource, a prioritized  
14 information processing unit change unit 180 for changing the  
15 prioritized information processing unit 150, a monitor control unit  
16 190 used when the degree of contention for the contended resource 50  
17 is relatively high, a fat lock storage unit 200, and a full stop  
18 unit 210 for temporarily stopping the operations of all the  
19 information processing units.

20 Each of the non-prioritized information processing units 160-1  
21 to 160-N executes, as an example of a first process, a process for  
22 writing on the flat lock storage unit 100 to the exclusion of the

1 other non-prioritized information processing units, thereby  
2 acquiring and releasing a non-prioritized exclusion right, which is  
3 a right to be a candidate for acquiring the contended resource 50.  
4 For example, each of the non-prioritized information processing  
5 units 160-1 to 160-N executes exclusive writes by a compare-and-swap  
6 instruction requiring a relatively long processing time. On the  
7 other hand, the prioritized information processing unit 150 set by  
8 the prioritized information processing unit setting unit 140  
9 executes, as an example of a second process requiring a shorter  
10 processing time than the first process, a process for writing on the  
11 flat lock storage unit 100 by, for example, a normal write  
12 instruction, thereby acquiring and releasing the contended resource  
13 50. The exclusion controller 10 allows any one of the prioritized  
14 information processing unit 150 and the non-prioritized information  
15 processing unit 160 which has acquired the non-prioritized exclusion  
16 right, to acquire the contended resource 50.

17 As described above, the exclusion controller 10 can allow the  
18 prioritized information processing unit 150 to acquire the contended  
19 resource 50 by using the second process faster than the first  
20 process. Accordingly, the exclusion controller 10 can allow the  
21 prioritized information processing unit 150 to acquire the contended  
22 resource 50 faster than the non-prioritized information processing

1 unit 160. For example, when it is previously predicted that the  
2 prioritized information processing unit 150 has a higher frequency  
3 of acquiring the contended resource 50 than the other information  
4 processing units, the exclusion controller 10 can execute an  
5 arbitration process regarding the acquisition of the contended  
6 resource 50 at a high speed.

7         The contended resource 50 is a resource to be exclusively  
8 acquired by any one of the prioritized information processing unit  
9 150 and the non-prioritized information processing units 160-1 to  
10 160-N, which asynchronously operate. For example, the contended  
11 resource 50 is a memory area which is shared by the prioritized  
12 information processing unit 150 and the non-prioritized information  
13 processing units 160-1 to 160-N and stores the total number of times  
14 that the resource has been acquired. The necessity of exclusively  
15 accessing this memory area will be described below.

16         The memory area can store the total number of times that the  
17 contended resource 50 has been acquired, in the case where the  
18 following increment process is executed when any one of the  
19 prioritized information processing unit 150 and the non-prioritized  
20 information processing units 160-1 to 160-N has acquired the  
21 contended resource: a counter is read from the memory area, a new  
22 value is generated by adding one to the read counter, and the new

1 value is written to the memory area.

2           However, if another information processing unit which has not  
3 acquired the contended resource is allowed to execute an increment  
4 process parallel to the above-described increment process, there may  
5 be cases where the memory area stores a value less than the total  
6 number of times that the contended resource 50 has been acquired.  
7 Specifically, if the another information processing unit initiates  
8 and terminates an increment process after the information processing  
9 unit, which has acquired the contended resource, has read the  
10 counter and before the information processing unit writes a new  
11 value, the result written by the another information processing unit  
12 is overwritten and destroyed by the new value generated by the  
13 information processing unit having acquired the contended resource.  
14 Therefore, in order to correctly keep the total number of times that  
15 the contended resource 50 has been acquired, the contended resource  
16 50 needs to be exclusively acquired by each of the plurality of  
17 information processing units.

18           Incidentally, instead of a memory area as described above, the  
19 contended resource 50 may be any one of the following: an  
20 input/output device provided in a computer or the like, a  
21 communication channel connected to a communication device, and an  
22 instruction sequence to be exclusively executed. In other words,

1 the contended resource 50 may be a device or the like which causes  
2 problems in operations when the device concurrently receives  
3 directions from a plurality of information processing units.

4 The flat lock storage unit 100 is provided so as to correspond  
5 to the contended source 50, and stores information indicating which  
6 information processing unit has acquired the contended resource 50.  
7 The flat lock storage unit 100 has a prioritized information  
8 processing unit information storage area 110, a prioritized  
9 exclusion right storage area 120, and a non-prioritized exclusion  
10 right storage area 130.

11 The prioritized information processing unit information  
12 storage area 110 stores prioritized information processing unit  
13 identification information indicating which information processing  
14 unit is the prioritized information processing unit 150. For  
15 example, the prioritized information processing unit information  
16 storage area 110 stores either anonymous state information  
17 indicating that any of the information processing units is not the  
18 prioritized information processing unit 150, or identification  
19 information for identifying the prioritized information processing  
20 unit 150.

21 The prioritized exclusion right storage area 120 stores  
22 "locked" as prioritized exclusion right information indicating that

1 the prioritized information processing unit 150 is trying to acquire  
2 the contended resource 50. On the other hand, when the prioritized  
3 information processing unit 150 is not trying to acquire the  
4 contended resource 50, the prioritized exclusion right storage area  
5 120 stores "unlocked."

6 The non-prioritized exclusion right storage area 130 stores,  
7 as non-prioritized exclusion right information identifying which of  
8 the non-prioritized information processing units 160-1 to 160-N has  
9 acquired the non-prioritized exclusion right, identification  
10 information for identifying the non-prioritized information  
11 processing unit 160. On the other hand, when any of the  
12 non-prioritized information processing units 160-1 to 160-N has not  
13 acquired the non-prioritized exclusion right, the non-prioritized  
14 exclusion right storage area 130 stores "nobody."

15 When the prioritized information processing unit setting unit  
16 140 receives, from any one of the prioritized information processing  
17 unit 150 and the non-prioritized information processing units 160-1  
18 to 160-N, directions to set the information processing unit as the  
19 prioritized information processing unit, the prioritized information  
20 processing unit setting unit 140 executes the following process to  
21 the exclusion of the other information processing units except the  
22 sender of the directions. If identification information for

1 identifying the prioritized information processing unit 150 is not  
2 stored in the prioritized information processing unit information  
3 storage area 110, that is, if anonymous state information is stored  
4 therein, the prioritized information processing unit setting unit  
5 140 stores, in the prioritized information processing unit  
6 information storage area 110, identification information for  
7 identifying the information processing unit which has given the  
8 directions to the prioritized information processing unit setting  
9 unit 140.

10         The non-prioritized information processing units 160-1 to  
11 160-N are, for example, units of processing which are managed by an  
12 operating system or a language processor and may operate  
13 asynchronously, specifically, threads or processes. As another  
14 example, each of the non-prioritized information processing units  
15 160-1 to 160-N may be a central processor of a computer.

16         When the non-prioritized information processing units 160-1 to  
17 160-N receive directions to acquire the contended resource 50 from a  
18 program or the like created by a user, the non-prioritized  
19 information processing units 160-1 to 160-N mutually exclusively  
20 acquire the non-prioritized exclusion right, which indicates a  
21 candidate for acquiring the contended resource 50, by using the  
22 first process. For example, the non-prioritized information

1 processing unit 160-1 executes the following process as the first  
2 process to the exclusion of the non-prioritized information  
3 processing units 160-2 to 160-N. If non-prioritized exclusion right  
4 information has not yet stored in the non-prioritized exclusion  
5 right storage area 130, that is, if "nobody" is stored therein, the  
6 non-prioritized information processing unit 160-1 writes  
7 non-prioritized exclusion right information indicating that the  
8 non-prioritized information processing unit 160-1 has acquired a  
9 non-prioritized exclusion right, in the non-prioritized exclusion  
10 right storage area 130.

11 To be more specific, the non-prioritized information  
12 processing unit 160-1 executes the first process by using an  
13 indivisible instruction (atomic instruction), such as a  
14 compare-and-swap instruction, which exclusively executes a read, a  
15 check, and a write without being interrupted by processes of the  
16 other information processing units. Instead of this, the  
17 non-prioritized information processing unit 160-1 may execute the  
18 first process by using any one of the following: a test-and-set  
19 instruction, a lock instruction which locks a bus used for writing  
20 on a memory, and an LL instruction or an SC instruction which is  
21 provided in, for example, a MIPS (trademark) processor and checks  
22 whether other information processing unit has written to a memory



1     having stored a read value.

2             Moreover, if an anonymous state is stored in the prioritized  
3     information processing unit information storage area 110, the  
4     non-prioritized information processing unit 160-1 transmits  
5     directions to set the non-prioritized information processing unit  
6     160-1 as the prioritized information processing unit, to the  
7     prioritized information processing unit setting unit 140.  
8     Operations of the non-prioritized information processing units 160-2  
9     to 160-N are almost the same as those of the non-prioritized  
10    information processing unit 160-1. Therefore, a description thereof  
11    will be omitted.

12            Further, when the non-prioritized information processing unit  
13    160-1 receives directions to release the contended resource 50, from  
14    a program or the like created by a user, the non-prioritized  
15    information processing unit 160-1 writes "nobody" in the  
16    non-prioritized exclusion right storage area 130, thereby releasing  
17    the contended resource 50. Note that, in transition to a fat state  
18    in which the contended resource 50 is acquired in a monitor mode to  
19    be described later, the non-prioritized information processing unit  
20    160-1 once acquires the contended resource 50 by using the monitor  
21    mode and then executes a process for releasing the contended  
22    resource 50 as needed.

1           Moreover, in the fat state, the non-prioritized information  
2 processing unit 160-1 acquires the contended resource 50 by  
3 transmitting directions to acquire the contended resource 50 to the  
4 monitor control unit 190. Similarly, in the fat state, the  
5 non-prioritized information processing unit 160-1 releases the  
6 contended resource 50 by transmitting directions to release the  
7 contended resource 50 to the monitor control unit 190.

8           Similar to the non-prioritized information processing units  
9 160-1 to 160-N, the prioritized information processing unit 150 is a  
10 unit of processing, such as a thread or a process. Realized  
11 examples of the prioritized information processing unit 150 are  
12 almost the same as those of the non-prioritized information  
13 processing unit 160. Therefore, a description thereof will be  
14 omitted.

15           The prioritized information processing unit 150 acquires the  
16 contended resource 50 by executing, as described below, the second  
17 process requiring a shorter processing time than the first process,  
18 to the exclusion of the non-prioritized information processing unit  
19 160 having acquired the non-prioritized exclusion right. First, the  
20 prioritized information processing unit 150 stores "locked" in the  
21 prioritized exclusion right storage area 120. Next, the prioritized  
22 information processing unit 150 reads the non-prioritized exclusion

1 right information stored in the non-prioritized exclusion right  
2 storage area 130. If non-prioritized exclusion right information  
3 has been already stored therein, that is, if any one of the  
4 non-prioritized information processing units 160-1 to 160-N has  
5 already acquired the non-prioritized exclusion right, the  
6 prioritized information processing unit 150 removes "locked" from  
7 the prioritized exclusion right storage area 120 to store "unlocked"  
8 therein, and then executes another process (e.g. a process for  
9 retrying to acquire the contended resource 50). On the other hand,  
10 if non-prioritized exclusion right information has not yet been  
11 stored, that is, if any of the non-prioritized information  
12 processing units 160-1 to 160-N has not acquired the non-prioritized  
13 exclusion right, the prioritized information processing unit 150  
14 acquires the contended resource 50 to execute a process using the  
15 contended resource 50.

16 When the prioritized information processing unit 150 receives  
17 directions to release the contended resource 50, from a program or  
18 the like created by a user, the prioritized information processing  
19 unit 150 releases the contended resource 50 by removing "locked"  
20 from the prioritized exclusion right storage area 120 to store  
21 "unlocked" therein. Note that a process for releasing the contended  
22 resource 50 in the fat state is almost the same as that of the

1 non-prioritized information processing unit 160-1 and therefore a  
2 description thereof will be omitted.

3         The acquisition check unit 170 checks whether the acquisition  
4 of the contended resource 50 has failed, based on the first or  
5 second process executed by an information processing unit. If  
6 "locked" is stored in the prioritized exclusion right storage area  
7 120, the acquisition check unit 170 checks whether the  
8 non-prioritized exclusion right is stored in the non-prioritized  
9 exclusion right storage area 130. If the acquisition check unit 170  
10 checks that the non-prioritized exclusion right is stored in the  
11 non-prioritized exclusion right storage area 130, the acquisition  
12 check unit 170 checks that the acquisition of the contended resource  
13 50 by the first process has failed. The acquisition check unit 170  
14 then transmits information indicating that the acquisition of the  
15 contended resource 50 has failed, to the prioritized information  
16 processing unit change unit 180 and the monitor control unit 190.

17         When the prioritized information processing unit change unit  
18 180 receives the information indicating that the acquisition of the  
19 contended resource 50 has failed from the acquisition check unit  
20 170, the prioritized information processing unit change unit 180  
21 stores anonymous state information in the prioritized information  
22 processing unit information storage area 110, thereby changing the

1 prioritized information processing unit 150 into the non-prioritized  
2 information processing unit 160. Moreover, when the prioritized  
3 information processing unit change unit 180 receives directions to  
4 change the prioritized information processing unit, from the full  
5 stop unit 210, the prioritized information processing unit change  
6 unit 180 specifies the information processing unit having acquired  
7 the contended resource 50 by referring to the flat lock storage unit  
8 100. Then, the prioritized information processing unit change unit  
9 180 writes, on the flat lock storage unit 100, information  
10 indicating that the specified information processing unit is set as  
11 the prioritized information processing unit, thereby changing the  
12 specified information processing unit into the prioritized  
13 information processing unit. The prioritized information processing  
14 unit change unit 180 determines the information processing unit to  
15 be changed into the prioritized information processing unit in the  
16 state where operations of the information processing units are  
17 stopped by the full stop unit 210. However, instead of this, the  
18 prioritized information processing unit change unit 180 may  
19 determine the information processing unit to be changed into the  
20 prioritized information processing unit by periodically referring to  
21 the flat lock storage unit 100 without stopping operations of the  
22 information processing units.

1           When the monitor control unit 190 receives the information  
2     indicating that the acquisition of the contended resource 50 has  
3     failed from the acquisition check unit 170, the monitor control unit  
4     190 executes a subsequent arbitration process regarding the  
5     acquisition of the contended resource 50 by using a monitor mode, an  
6     example of which will be described below. In the monitor mode, the  
7     fat lock storage unit 200 has recorded identification information of  
8     all the information processing units which are waiting for the  
9     contended resource 50 to be released by another information  
10    processing unit in order to acquire the contended resource 50.  
11    Specifically, when the monitor control unit 190 receives directions  
12    to acquire the contended resource 50, from any one of the  
13    prioritized information processing unit 150 and the non-prioritized  
14    information processing units 160-1 to 160-N, the monitor control  
15    unit 190 registers information for identifying the information  
16    processing unit having transmitted the directions, as the  
17    identification information of the information processing unit  
18    waiting for the release of the contended resource 50, in the fat  
19    lock storage unit 200. The exclusion controller 10 does not  
20    allocate a computational resource, such as a CPU, to these waiting  
21    information processing units. Accordingly, the entire system is in  
22    a state where the computational resource is efficiently utilized.

1           When the monitor control unit 190 receives directions to  
2   release the contended resource 50, from an information processing  
3   unit, the monitor control unit 190 selects the information  
4   processing unit which satisfies a predetermined condition, for  
5   example, the information processing unit which has first started  
6   waiting, among the information processing units waiting in order to  
7   acquire the contended resource 50, thus allowing the selected  
8   information processing unit to acquire the contended resource 50.

9           Incidentally, when the monitor control unit 190 receives the  
10   information indicating that the acquisition of the contended  
11   resource 50 has failed from the acquisition check unit 170, the  
12   monitor control unit 190 may execute a subsequent arbitration  
13   process regarding the acquisition of the contended resource 50 by  
14   using another arbitrary locking method instead of the  
15   above-described process. For example, a locking method which is  
16   provided as standard in the operating system managing the exclusion  
17   controller 10 may be used.

18           As described above, when the monitor control unit 190 allows  
19   an information processing unit to try to acquire the contended  
20   resource 50, the monitor control unit 190 registers a waiting state  
21   and the like in the fat lock storage unit 200. Accordingly, a long  
22   processing time is required for completing an acquisition check

1 after acquisition is attempted, compared to the acquisition of the  
2 contended resource 50 by using the flat lock storage unit 100.  
3 However, when there are relatively many information processing units  
4 which are waiting in order to acquire the contended resource 50, the  
5 monitor control unit 190 does not allocate the computational  
6 resource to these waiting information processing units, thereby  
7 improving the utilization ratio of the computational resource in the  
8 whole system.

9 In the subsequent description, the acquisition of the  
10 contended resource 50 using the flat lock storage unit 100 is  
11 referred to as flat lock, and the acquisition of the contended  
12 resource 50 using the fat lock storage unit 200 is referred to as  
13 fat lock, as names according to the above-described characteristics.  
14 Moreover, a state where all the information processing units use the  
15 fat lock in order to acquire the contended resource 50 is referred  
16 to as a fat state or a weighing mode, and a state where all the  
17 information processing units use the flat lock in order to acquire  
18 the contended resource 50 is referred to as a flat state or a  
19 priority mode.

20 Incidentally, a locking method used in the fat state is not  
21 limited to the aforementioned monitor mode. If the degree of  
22 contention is higher than that in the acquisition of the contended



1 resource 50 by using the flat lock storage unit 100 by a  
2 predetermined degree, for example, if the frequency of contention is  
3 a predetermined value or more, the monitor control unit 190 may  
4 allow an information processing unit to acquire the contended  
5 resource 50 by using a method of faster operations.

6         The fat lock storage unit 200 acquires data necessary for the  
7 monitor mode used by the monitor control unit 190. The fat lock  
8 storage unit 200 typically stores information for identifying the  
9 information processing unit having acquired the contended resource  
10 50 and information indicating which of the information processing  
11 units are waiting in order to acquire the contended resource 50.  
12 For example, the fat lock storage unit 200 may manage the  
13 information indicating the information processing units waiting in  
14 order to acquire the contended resource 50 in the order of wait  
15 initiation time by connecting the information to a first-in  
16 first-out (FIFO) queue. In this case, the fat lock storage unit 200  
17 may remove the information indicating an information processing unit  
18 from the queue or may newly add the information indicating an  
19 information processing unit to the queue in accordance with  
20 directions from the monitor control unit 190.

21         The full stop unit 210 periodically stops operations of the  
22 prioritized information processing unit 150 and the non-prioritized

1 information processing units 160-1 to 160-N. For example, when the  
2 exclusion controller 10 uses a garbage collection (abbreviated to  
3 GC) function to manage a memory, the full stop unit 210 periodically  
4 stops operations of the prioritized information processing unit 150  
5 and the non-prioritized information processing units 160-1 to 160-N  
6 in order to execute stop-the-world GC. In addition to executing GC,  
7 the full stop unit 210 transmits directions to change the  
8 prioritized information processing unit to the prioritized  
9 information processing unit change unit 180.

10 As described above, the exclusion controller 10 can allow the  
11 prioritized information processing unit 150 to acquire the contended  
12 resource 50 at a high speed. Accordingly, if the prioritized  
13 information processing unit 150 has a higher frequency of acquiring  
14 the contended resource 50 than the other information processing  
15 units, an arbitration process regarding the acquisition of the  
16 contended resource 50 can be executed at a high speed.

17 Figs. 2A to 2E show details of the flat lock storage unit 100  
18 in the first embodiment. The flat lock storage unit 100 includes  
19 the prioritized information processing unit information storage area  
20 110 for storing information for identifying the prioritized  
21 information processing unit, the prioritized exclusion right storage  
22 area 120 for storing prioritized exclusion right information, and

1 the non-prioritized exclusion right storage area 130 for storing  
2 non-prioritized exclusion right information. The information for  
3 identifying the prioritized information processing unit and the  
4 non-prioritized exclusion right information may be, for example,  
5 thread IDs of information processing units which are managed by an  
6 operating system, or may be thread IDs of information processing  
7 units which are managed by a language processor for operating the  
8 exclusion controller 10. Moreover, the prioritized exclusion right  
9 information is, for example, binary information indicating whether  
10 the prioritized information processing unit 150 is trying to acquire  
11 the contended resource 50.

12 Figure 2A shows an example of the flat lock storage unit 100  
13 at the time when the contended resource 50 is initialized. The  
14 prioritized information processing unit information storage area 110  
15 stores "anonymous state" indicating that none of the information  
16 processing units is the prioritized information processing unit 150.  
17 Moreover, the prioritized exclusion right storage area 120 stores  
18 "unlocked" indicating that the prioritized information processing  
19 unit 150 is not trying to acquire the contended resource 50.  
20 Further, the non-prioritized exclusion right storage area 130 stores  
21 "nobody" indicating that none of the information processing units  
22 has acquired the non-prioritized exclusion right.

1           Figure 2B shows the state where the prioritized information  
2     processing unit 150 has been set by the prioritized information  
3     processing unit setting unit 140 in Figure 2A. The prioritized  
4     information processing unit information storage area 110 stores ID1,  
5     which is information for identifying the prioritized information  
6     processing unit 150. Figure 2C shows the state where the  
7     acquisition of the contended resource 50 has been tried by the  
8     prioritized information processing unit 150 in Figure 2B. The  
9     prioritized exclusion right storage area 120 stores "locked"  
10    indicating that the acquisition of the contended resource 50 has  
11    been tried by the prioritized information processing unit 150.  
12    Figure 2D shows the state where the non-prioritized exclusion right  
13    has been acquired by the non-prioritized information processing unit  
14    160-1 in Figure 2B. The non-prioritized exclusion right storage  
15    area 130 stores ID2. This ID2 indicates that the non-prioritized  
16    exclusion right has been acquired by the information processing unit  
17    identified by the identification information of ID2, e.g. the  
18    non-prioritized information processing unit 160-1.

19           Figure 2E shows the state where the acquisition of the  
20    contended resource 50 has been tried by the prioritized information  
21    processing unit 150 in Figure 2D, and also the state where the  
22    non-prioritized exclusion right has been acquired by the

1 non-prioritized information processing unit 160-1 in the state shown  
2 in Figure 2C. The prioritized exclusion right storage area 120  
3 stores "locked," and the non-prioritized exclusion right storage  
4 area 130 stores "ID2" indicating that the non-prioritized exclusion  
5 right has been acquired by the non-prioritized information  
6 processing unit 160-1. Two cases leading to the state of Figure 2E  
7 will be described in order.

8 In the state shown in Figure 2D, the prioritized information  
9 processing unit 150 stores "locked" in the prioritized exclusion  
10 right storage area 120 in order to acquire the contended resource 50  
11 in response to directions or the like from a user program. The  
12 state at this point is shown in Figure 2E. Thereafter, the  
13 prioritized information processing unit 150 reads non-prioritized  
14 exclusion right information stored in the non-prioritized exclusion  
15 right storage area 130. The prioritized information processing unit  
16 150 checks that non-prioritized exclusion right information has been  
17 already stored therein, i.e. that the non-prioritized information  
18 processing unit 160-1 has already acquired the non-prioritized  
19 exclusion right. The prioritized information processing unit 150  
20 then removes "locked" from the prioritized exclusion right storage  
21 area 120 to store "unlocked" therein, thereby canceling an attempt  
22 to acquire the contended resource 50 (return to the state of Figure

1 2D).

2 In the state shown in Figure 2C, when the non-prioritized  
3 information processing unit 160-1 receives directions to acquire the  
4 contended resource 50, from a user program or the like, the  
5 non-prioritized information processing unit 160-1 stores "ID2" in  
6 the non-prioritized exclusion right storage area 130 to the  
7 exclusion of the non-prioritized information processing units 160-2  
8 to 160-N. The state at this point is shown in Figure 2E.  
9 Thereafter, the non-prioritized information processing unit 160-1  
10 removes "ID2" stored in the non-prioritized exclusion right storage  
11 area 130, thereby releasing the non-prioritized exclusion right  
12 (return to the state of Figure 2C).

13 As described above, the flat lock storage unit 100 has the  
14 prioritized exclusion right storage area 120, in which only the  
15 prioritized information processing unit 150 writes, and the  
16 non-prioritized exclusion right storage area 130, in which the  
17 non-prioritized information processing units 160-1 to 160-N write,  
18 individually. Therefore, since the prioritized information  
19 processing unit 150 need not write in the prioritized exclusion  
20 right storage area 120 to the exclusion of the other information  
21 processing units, the prioritized information processing unit 150  
22 can operate at a high speed.

1           Figure 3 shows an operation flow where the prioritized  
2 information processing unit 150 or the non-prioritized information  
3 processing unit 160 tries to acquire the contended resource in the  
4 first embodiment. The prioritized information processing unit 150  
5 and the non-prioritized information processing units 160-1 to 160-N  
6 can concurrently operate in accordance with the operation flows  
7 shown in Figs. 3 to 6. Moreover, in Figure 3, the prioritized  
8 information processing unit 150 and the non-prioritized information  
9 processing units 160-1 to 160-N operate in almost the same manner.  
10 Therefore, the prioritized information processing unit 150 and the  
11 non-prioritized information processing units 160-1 to 160-N are  
12 generically named information processing units in the description  
13 below.

14           An information processing unit tries to acquire the flat lock  
15 by using the first process and/or the second process in order to  
16 acquire the contended resource 50 (S100A). Details thereof will be  
17 described later. If the flat lock has been successfully acquired  
18 (S110: YES), the information processing unit acquires the contended  
19 resource 50 to execute a process using the contended resource 50  
20 (S120).

21           On the other hand, if the acquisition check unit 170 has  
22 failed to acquire the flat lock (S110: NO), the monitor control unit

1 190 allows the information processing unit to wait until the  
2 information processing unit acquires the fat lock (S130). After the  
3 information processing unit has acquired the fat lock, the  
4 information processing unit checks whether the transition to the fat  
5 state has been completed (S140). Details of the check as to whether  
6 the transition to the fat state has been completed will be described  
7 later in the description of S170. If the information processing  
8 unit has checked that the transition to the fat state has been  
9 completed (S140: YES), the information processing unit acquires the  
10 contended resource 50 by the fat lock to execute a process using the  
11 contended resource 50 (S120).

12 On the other hand, if the information processing unit has  
13 checked that the transition to the fat state has not yet been  
14 completed (S140: NO), the information processing unit sets the  
15 contended resource 50 to be in a flat lock contention state, which  
16 indicates that the acquisition of the flat lock has failed and the  
17 transition to the fat state has not been completed (S150). For  
18 example, the information processing unit writes information  
19 indicating the flat lock contention state in a predetermined area on  
20 a memory accessible by all the information processing units, thereby  
21 setting the contended resource 50 to be in the flat lock contention  
22 state.



1           Subsequently, the information processing unit tries to acquire  
2 the flat lock by using the first process and/or the second process  
3 (S100B). If the flat lock has been successfully acquired (S160:  
4 YES), the information processing unit sets the contended resource 50  
5 to be in the fat state (S170), and returns to S140. Details of S170  
6 will be described later.

7           On the other hand, if the acquisition of the flat lock has  
8 failed (S160: NO), the information processing unit releases the fat  
9 lock already acquired in S130 to wait until the information  
10 processing unit receives a lock state change notice from another  
11 information processing unit (S180). Thereafter, if the information  
12 processing unit has received a lock state change notice from another  
13 information processing unit, the information processing unit again  
14 acquires the fat lock to return to the process of S140.

15           As described above, the exclusion controller 10 allows the  
16 information processing unit having acquired both the flat lock and  
17 the fat lock to execute the transition process from the flat state  
18 to the fat state. Accordingly, the exclusion controller 10 can  
19 avoid disorder or the like in which both the flat lock and the fat  
20 lock coexist as locks for acquiring the contended resource 50, in  
21 the transition process from the flat lock to the fat lock.

22           Figure 4 is an operation flow showing details of S100A and

1 S100B in Figure 3. If the information processing unit executing  
2 S100 checks that the information processing unit is the prioritized  
3 information processing unit 150 (S200: YES), the information  
4 processing unit stores "locked" in the prioritized exclusion right  
5 storage area 120 (S210). Then, if non-prioritized exclusion right  
6 information is not stored in the non-prioritized exclusion right  
7 storage area 130, that is, if "nobody" is stored therein (S220: NO),  
8 the prioritized information processing unit 150 acquires the  
9 contended resource 50 (S240), and the process comes to an end.

10 On the other hand, if non-prioritized exclusion right  
11 information has been already stored in the non-prioritized exclusion  
12 right storage area 130 (S220: YES), that is, if any one of the  
13 non-prioritized information processing units 160-1 to 160-N has  
14 already acquired the non-prioritized exclusion right, the  
15 acquisition check unit 170 removes "locked" stored in the  
16 prioritized exclusion right storage area 120 (S250). The  
17 prioritized information processing unit change unit 180 then sets  
18 prioritized information processing unit information to an anonymous  
19 state by storing anonymous state information in the prioritized  
20 information processing unit information storage area 110 (S260), and  
21 does not allow the prioritized information processing unit 150 to  
22 acquire the contended resource 50 (S270), and then the process comes

1 to an end.

2           On the other hand, if the information processing unit  
3       executing S100 has checked that the information processing unit is  
4       any one of the non-prioritized information processing units, e.g.  
5       the non-prioritized information processing unit 160-1 (S200: NO),  
6       the information processing unit checks whether the prioritized  
7       information processing unit information stored in the prioritized  
8       information processing unit information storage area 110 indicates  
9       an anonymous state (S280). If the prioritized information  
10      processing unit information indicates an anonymous state (S280:  
11      YES), the prioritized information processing unit setting unit 140  
12      executes a process for setting the non-prioritized information  
13      processing unit 160-1 as the prioritized information processing unit  
14      to the exclusion of the other non-prioritized information processing  
15      units, that is, a process for writing, in the prioritized  
16      information processing unit information storage area 110, the  
17      prioritized information processing unit information indicating that  
18      the non-prioritized information processing unit 160-1 is the  
19      prioritized information processing unit (S290), and then the process  
20      returns to S200. In other words, the prioritized information  
21      processing unit setting unit 140 sets as the prioritized information  
22      processing unit 150 the information processing unit having first

1     acquired the contended resource 50 after initialization thereof, and  
2     sets as the non-prioritized information processing units 160 the  
3     other information processing units except the prioritized  
4     information processing unit 150.

5             If the prioritized information processing unit information is  
6     not in an anonymous state (S280: NO), the non-prioritized  
7     information processing unit 160-1 executes a process for acquiring  
8     the non-prioritized exclusion right to the exclusion of the other  
9     non-prioritized information processing units (S300).

10            If the acquisition of the non-prioritized exclusion right has  
11     failed (S310: NO), the non-prioritized information processing unit  
12     160-1 cannot acquire the contended resource 50 (S270), and the  
13     process comes to an end. On the other hand, if the non-prioritized  
14     exclusion right has been successfully acquired (S310: YES), the  
15     non-prioritized information processing unit 160-1 checks whether  
16     "locked" has been already stored in the prioritized exclusion right  
17     storage area 120 (S320). If "locked" is not stored therein (S320:  
18     NO), the non-prioritized information processing unit 160-1 acquires  
19     the contended resource 50 (S240), and the process comes to an end.  
20     On the other hand, if "locked" is stored therein (S320: YES), the  
21     non-prioritized information processing unit 160-1 removes the  
22     information indicating the non-prioritized exclusion right from the

1 prioritized exclusion right storage area 120 (S330). In this case,  
2 the non-prioritized information processing unit 160-1 cannot acquire  
3 the contended resource 50 (S270), and the process comes to an end.

4 As described above, the non-prioritized information processing  
5 unit 160-1 executes, as the first process, a process (e.g. S200,  
6 S280, S300, S310, S320, and S240) for storing non-prioritized  
7 exclusion right information indicating that the non-prioritized  
8 information processing unit has acquired the non-prioritized  
9 exclusion right, to the exclusion of the other non-prioritized  
10 information processing units if non-prioritized exclusion right  
11 information has not been stored. On the other hand, the prioritized  
12 information processing unit 150 can acquire the contended resource  
13 50 by executing, as the second process requiring a shorter  
14 processing time than the first process, a process (e.g. S200, S210,  
15 S220, and S240) for storing "locked" in the prioritized exclusion  
16 right storage area 120.

17 Figure 5 shows an operation flow showing details of S170 in  
18 Figure 3. The information processing unit resets the flat lock  
19 contention state (S400). Next, the information processing unit  
20 notifies all the information processing units waiting for a lock  
21 state change notice to arrive (e.g. S180 of Figure 3) of the change  
22 of the lock state (S410). Then, the information processing unit

1 sets the contended resource 50 to the fat state (S420).

2 The process for setting the contended resource 50 to the fat  
3 state is, for example, a process for writing information indicating  
4 the fat state in a predetermined area (e.g. the flat lock storage  
5 unit 100) on a memory which is accessible by all the information  
6 processing units capable of acquiring the contended resource 50 and  
7 is provided so as to correspond to the contended resource 50.  
8 Therefore, the other information processing units can check whether  
9 the contended resource 50 is currently in the fat state, by  
10 referring to the predetermined area (e.g. S140 of Figure 3).

11 Figure 6 shows an operation flow where the prioritized  
12 information processing unit 150 or the non-prioritized information  
13 processing unit 160 releases the contended resource 50. First, the  
14 information processing unit checks whether the contended resource 50  
15 is in the flat state, by referring to the flat lock storage unit 100  
16 (S500). If the information processing unit has checked that the  
17 contended resource 50 is not in the flat state (S500: NO), the  
18 information processing unit checks whether a condition for the  
19 transition from the fat state to the flat state is satisfied, e.g.,  
20 whether there is an information processing unit waiting in order to  
21 acquire the contended resource 50 (S510).

22 If the condition for the transition from the fat state to the

1 flat state is not satisfied (S510: NO), the information processing  
2 unit executes a process for releasing the fat lock by using the  
3 monitor control unit 190 (S520). On the other hand, if the  
4 condition for the transition from the fat state to the flat state is  
5 satisfied (S510: YES), the information processing unit initializes  
6 the flat lock storage unit 100 (S530), and sets the state for  
7 acquiring the contended resource 50 to the flat state (S540).

8 Incidentally, when the information processing unit initializes  
9 the flat lock storage unit 100, the prioritized information  
10 processing unit may be set to be in an anonymous state. In this  
11 case, the information processing unit can set, as the prioritized  
12 information processing unit 150, the information processing unit  
13 first acquiring the contended resource 50 after the transition to  
14 the flat lock.

15 On the other hand, if the information processing unit has  
16 checked that the contended resource 50 is in the flat state (S500:  
17 YES), the information processing unit executes a process for  
18 releasing the flat lock, e.g., a process for storing information  
19 indicating that the information processing unit has released the  
20 contended resource 50, in the prioritized exclusion right storage  
21 area 120 or the non-prioritized exclusion right storage area 130  
22 (S550). Subsequently, the information processing unit checks

1 whether the contended resource 50 is set to the flat lock contention  
2 state (S560). If the contended resource 50 is not set to be in the  
3 flat lock contention state (S560: NO), the releasing process comes  
4 to an end. On the other hand, if the contended resource 50 is set  
5 to be in the flat lock contention state, the information processing  
6 unit tries to acquire the fat lock. Then, if the information  
7 processing unit has acquired the fat lock (S570: YES), the  
8 information processing unit again checks whether the contended  
9 resource 50 is set to be in the flat lock contention state (S580).  
10 If the contended resource 50 is set to be in the flat lock  
11 contention state (S580: YES), the information processing unit  
12 notifies any one of the other information processing units waiting  
13 for a lock state change notice to arrive, of the change of the lock  
14 state (S590). Subsequently, the information processing unit  
15 releases the fat lock (S520).

16 As described above, if the contended resource 50 is in the  
17 flat lock contention state, which is the transition process from the  
18 flat lock to the fat lock, the exclusion controller 10 once acquires  
19 both the flat lock and the fat lock and then releases both the  
20 locks. Therefore, even in a transient state where both the flat  
21 lock and the fat lock coexist as locks for acquiring the contended  
22 resource 50, the contended resource 50 can be appropriately



1 released.

2 (Modified Example)

3 Figure 7 shows a function block diagram of an exclusion  
4 controller 10 in a modified example of the first embodiment. Since  
5 the exclusion controller 10 in Figure 7 has almost the same  
6 configuration as that of the exclusion controller 10 shown in Figure  
7 1, only the differences therebetween will be described. The  
8 exclusion controller 10 of the present modified example need not  
9 include a prioritized information processing unit setting unit 140,  
10 a prioritized information processing unit change unit 180, a monitor  
11 control unit 190, a flat lock storage unit 200, and a full stop unit  
12 210, unlike the exclusion controller 10 shown in Figure 1.  
13 Moreover, the flat lock storage unit 100 in Figure 7 may have an  
14 area usage type storage area 115, an exclusion right storage area  
15 125, and an information processing unit identification information  
16 storage area 135, instead of the prioritized information processing  
17 unit information storage area 110, the prioritized exclusion right  
18 storage area 120, and the non-prioritized exclusion right storage  
19 area 130 in the flat lock storage unit 100 of Figure 1,  
20 respectively. Here, the exclusion right storage area 125 is an  
21 example of a prioritized exclusion right storage area according to  
22 the present invention, and the information processing unit

1 identification information storage area 135 is an example of a  
2 non-prioritized exclusion right storage area according to the  
3 present invention.

4 The information processing unit identification information  
5 storage area 135 stores any one of prioritized information  
6 processing unit information and non-prioritized exclusion right  
7 information. The area usage type storage area 115 stores area usage  
8 type information indicating which of the prioritized information  
9 processing unit information and the non-prioritized exclusion right  
10 information the information processing unit identification  
11 information storage area 135 stores. Note that, since the area  
12 usage type information is binary information, the area usage type  
13 storage area 115 has a smaller size than that of the prioritized  
14 information processing unit information storage area 110 in Figure  
15 1.

16 The prioritized information processing unit 150 acquires the  
17 contended resource 50 by storing "locked" in the exclusion right  
18 storage area 125, and releases the contended resource 50 by removing  
19 "locked" from the exclusion right storage area 125. If the  
20 information processing unit identification information storage area  
21 135 does not store the identification information of the prioritized  
22 information processing unit 150 in the case where the prioritized

1 information processing unit 150 has stored "locked" in the exclusion  
2 right storage area 125, the acquisition check unit 170 writes  
3 "unlocked" in the exclusion right storage area 125, thereby  
4 canceling the acquisition of the contended resource 50 by the  
5 prioritized information processing unit 150.

6 The non-prioritized information processing unit 160 acquires  
7 an access right to the flat lock storage unit 100 to the exclusion  
8 of the other information processing units, and executes the  
9 following process. If "locked" is not stored in the exclusion right  
10 storage area 125, the non-prioritized information processing unit  
11 160 stores the identification information of the non-prioritized  
12 information processing unit 160 in the information processing unit  
13 identification information storage area 135, and stores "locked" in  
14 the exclusion right storage area 125. Further, the non-prioritized  
15 information processing unit 160 stores the area usage type  
16 information indicating that non-prioritized exclusion right  
17 information is stored, in the area usage type storage area 115.

18 As described above, if only the prioritized information  
19 processing unit 150 acquires the contended resource 50, the  
20 prioritized information processing unit 150 acquires the contended  
21 resource 50 at a high speed by writing in the exclusion right  
22 storage area 125. On the other hand, once the non-prioritized

1 information processing unit 160 succeeds in acquiring the contended  
2 resource 50, the non-prioritized information processing unit 160  
3 allows the information processing unit identification information  
4 storage area 135 to store non-prioritized exclusion right  
5 information, thereby acquiring the contended resource 50 to the  
6 exclusion of the other non-prioritized information processing units.  
7 In this case, the prioritized information processing unit 150 loses  
8 an originally possessed priority right to acquire the contended  
9 resource 50, and acquires the contended resource 50 by acquiring the  
10 non-prioritized exclusion right similarly to the non-prioritized  
11 information processing unit 160.

12 Figs. 8A to 8D show details of the flat lock storage unit 100  
13 in the modified example of the first embodiment. The flat lock  
14 storage unit 100 includes the area usage type storage area 115 for  
15 storing an area usage type, the exclusion right storage area 125 for  
16 storing prioritized exclusion right information, and the information  
17 processing unit identification information storage area 135 for  
18 storing non-prioritized exclusion right information. The area usage  
19 type information is binary information indicating which of the  
20 prioritized information processing unit information and the  
21 non-prioritized exclusion right information the information  
22 processing unit identification information storage area 135 stores.

1 In other words, the size of the area usage type storage area 115 is,  
2 for example, one bit.

3 Figure 8A shows an example of the flat lock storage unit 100  
4 at the time when the contended resource 50 is initialized. The area  
5 usage type storage area 115 stores "prioritized information  
6 processing unit" indicating that the information processing unit  
7 identification information storage area 135 stores prioritized  
8 information processing unit information. The exclusion right  
9 storage area 125 stores "unlocked" indicating that the prioritized  
10 information processing unit 150 is not trying to acquire the  
11 contended resource 50. The information processing unit  
12 identification information storage area 135 stores "ID1" as  
13 information for identifying the prioritized information processing  
14 unit 150.

15 Figure 8B shows the state where the prioritized information  
16 processing unit 150 has tried to acquire the contended resource 50  
17 in Figure 8A. The exclusion right storage area 125 stores "locked"  
18 indicating that the prioritized information processing unit 150 has  
19 tried to acquired the contended resource 50.

20 Figure 8C shows the state where the non-prioritized  
21 information processing unit 160-1 has acquired the non-prioritized  
22 exclusion right in Figure 8A. The area usage type storage area 115

1 stores "non-prioritized exclusion right" indicating that the  
2 information processing unit identification information storage area  
3 135 stores non-prioritized exclusion right information. The  
4 information processing unit identification information storage area  
5 135 stores ID2 as information for identifying the non-prioritized  
6 information processing unit 160-1, in order to indicate that the  
7 non-prioritized exclusion right has been acquired by the  
8 non-prioritized information processing unit 160-1.

9 Here, in order to change the state of Figure 8A into the state  
10 of Figure 8C, the non-prioritized information processing unit 160-1  
11 executes the following process to the exclusion of the other  
12 information processing units. First, the non-prioritized  
13 information processing unit 160-1 checks whether "nobody" indicating  
14 that none of the information processing units has acquired the  
15 contended resource 50 is stored in the information processing unit  
16 identification information storage area 135, and at the same time,  
17 whether "unlocked" is stored in the exclusion right storage area  
18 125. Then, if both results are YES, ID2 is stored in the  
19 information processing unit identification information storage area  
20 135, and "locked" is stored in the exclusion right storage area 125.  
21 It is desirable that this series of processes are executed by using  
22 a compare-and-swap instruction or the like to the exclusion of the

1 other information processing units. Even if a compare-and-swap  
2 instruction is applicable only to one word on memory, the  
3 non-prioritized information processing unit 160-1 can appropriately  
4 execute a process for acquiring the non-prioritized exclusion right  
5 by setting the exclusion right storage area 125 and the information  
6 processing unit identification information storage area 135 as  
7 continuous areas as shown in the present embodiment.

8         Figure 8D shows the state where the non-prioritized exclusion  
9 right has been released by the non-prioritized information  
10 processing unit 160-1 in Figure 8C. The information processing unit  
11 identification information storage area 135 stores "nobody."

12         As described above, the flat lock storage unit 100 in the  
13 present example has a small size compared to the flat lock storage  
14 unit 100 in Figure 1. Accordingly, the exclusion controller 10 of  
15 the present modified example can appropriately arbitrate the  
16 acquisition of the contended resource 50 by using a smaller storage  
17 area than that of the exclusion controller 10 in Figure 1, while  
18 allowing the prioritized information processing unit 150 to operate  
19 at a higher speed than the non-prioritized information processing  
20 unit 160 similarly to Figure 1.

21         Figure 9 is an operation flow where the prioritized  
22 information processing unit 150 or the non-prioritized information

1 processing unit 160 tries to acquire the contended resource in the  
2 modified example of the first embodiment. In some parts of this  
3 flow, both the prioritized information processing unit 150 and the  
4 non-prioritized information processing unit 160 operate in almost  
5 the same manner. Accordingly, in such parts, the prioritized  
6 information processing unit 150 and the non-prioritized information  
7 processing unit 160 are generically referred to as information  
8 processing units.

9       When an information processing unit tries to acquire the  
10 contended resource 50, the information processing unit checks  
11 whether the information processing unit identification information  
12 storage area 135 stores the ID for identifying the information  
13 processing unit (S600). If the information processing unit  
14 identification information storage area 135 stores the ID for  
15 identifying the information processing unit (S600: YES), the  
16 prioritized information processing unit 150 stores "locked" in the  
17 exclusion right storage area 125. The prioritized information  
18 processing unit 150 then reads the information processing unit  
19 identification information storage area 135 again. If the  
20 prioritized information processing unit 150 has checked that the  
21 information in the information processing unit identification  
22 information storage area 135 has not changed from the information at



1 the time of S600 (S620: YES), the contended resource 50 is  
2 successfully acquired (S630), and the process comes to an end. On  
3 the other hand, if the prioritized information processing unit 150  
4 has checked that the information in the information processing unit  
5 identification information storage area 135 has changed from the  
6 information at the time of S600 (S620: NO), the prioritized  
7 information processing unit 150 removes "locked" from the exclusion  
8 right storage area 125 to store "unlocked" therein. In this case,  
9 the prioritized information processing unit 150 fails to acquire the  
10 contended resource 50 (S650), and the process comes to an end.

11 If the information processing unit identification information  
12 storage area 135 does not store the ID for identifying the  
13 information processing unit (S600: NO), the non-prioritized  
14 information processing unit 160 checks whether an ID other than  
15 "nobody" is stored in the information processing unit identification  
16 information storage area 135 (S660). If an ID other than "nobody"  
17 is stored in the information processing unit identification  
18 information storage area 135 (S660: YES), the non-prioritized  
19 information processing unit 160 tries to acquire the non-prioritized  
20 exclusion right to the exclusion of the other non-prioritized  
21 information processing units (S670). If the acquisition has failed  
22 (S680: NO), the information processing unit fails to acquire the

1     contended resource 50 (S650), and the process comes to an end. On  
2     the other hand, if the non-prioritized exclusion right has been  
3     successfully acquired (S680: YES), the information processing unit  
4     stores, in the area usage type storage area 115, information  
5     indicating that the information processing unit identification  
6     information storage area 135 stores non-prioritized exclusion right  
7     information (S690). In this case, the information processing unit  
8     succeeds in acquiring the contested resource 50 (S630), and the  
9     process comes to an end.

10           If "nobody" is stored in the information processing unit  
11     identification information storage area 135 (S660: NO), the  
12     non-prioritized information processing unit 160 tries to acquire the  
13     non-prioritized exclusion right to the exclusion of the other  
14     non-prioritized information processing units (S700). If the  
15     acquisition has failed (S710: NO), the non-prioritized information  
16     processing unit 160 fails to acquire the contested resource 50  
17     (S730), and the process comes to an end. On the other hand, if the  
18     non-prioritized exclusion right has been successfully acquired  
19     (S710: YES), the non-prioritized information processing unit 160  
20     succeeds in acquiring the contested resource 50 (S720), and the  
21     process comes to an end. The information processing unit may  
22     operate by using a spin lock in which the above-described operations

1 are repeated until the contended resource 50 is successfully  
2 acquired, or may execute a process for transitioning to another  
3 locking method if the acquisition of the contended resource 50 has  
4 failed.

5 As described above, when only the prioritized information  
6 processing unit 150 has acquired the contended resource 50, the  
7 exclusion controller 10 allows the prioritized information  
8 processing unit 150 to acquire the contended resource 50 by writing  
9 on the flat lock storage unit 100, thus enabling fast operations.  
10 On the other hand, even when the non-prioritized information  
11 processing unit 160 has acquired the contended resource 50, the  
12 exclusion controller 10 can appropriately arbitrate the acquisition  
13 of the contended resource 50.

14 (Second Embodiment)

15 Figure 10 is a function block diagram of an exclusion  
16 controller 20 in a second embodiment. The exclusion controller 20  
17 includes a contended resource 60, a resource information storage  
18 area 300, a prioritized information processing unit 340 as an  
19 example of a first thread, non-prioritized information processing  
20 units 350-1 to 350-N as examples of a second thread, an execution  
21 state acquisition/notification unit 390, and a monitor control unit  
22 410. The exclusion controller 20 has an object to arbitrate the

1 acquisition of the contended resource 60 among the prioritized  
2 information processing unit 340 and the non-prioritized information  
3 processing units 350-1 to 350-N.

4         The contended resource 60 is a resource to be exclusively  
5 acquired by any one of the prioritized information processing unit  
6 340 and the non-prioritized information processing units 350-1 to  
7 350-N, which asynchronously operate. Other details of the contended  
8 resource 60 are almost the same as those of the contended resource  
9 50 described in Figure 1. Therefore, a description thereof will be  
10 omitted.

11         The resource information storage area 300 has a priority right  
12 information field 310, a recursive acquisition phase field 320, and  
13 a mode type field 330. The priority right information field 310  
14 stores priority right information indicating that the prioritized  
15 information processing unit exists. For example, the priority right  
16 information is information (thread ID or the like) for identifying  
17 the prioritized information processing unit. If no information  
18 processing unit is set as the prioritized information processing  
19 unit, the priority right information field 310 stores anonymous  
20 state information indicating that no information processing unit is  
21 set as the prioritized information processing unit. The recursive  
22 acquisition phase field 320 stores resource information indicating

1 whether the contended resource 60 has been acquired by any one of  
2 the plurality of information processing units. The mode type field  
3 330 stores mode type information identifying either a priority mode  
4 or a normal mode. In the priority mode, the resource information  
5 storage area 300 is used for acquiring the contended resource 60.  
6 In the normal mode, the monitor control unit 410 is used for  
7 acquiring the contended resource 60.

8         If the mode type information stored in the mode type field 330  
9 indicates the priority mode, the prioritized information processing  
10 unit 340 executes the following process. First, the prioritized  
11 information processing unit 340 executes, as an example of the  
12 second process according to the present invention, a process for  
13 reading the contents of the priority right information field 310 and  
14 writing resource information in the recursive acquisition phase  
15 field 320 if priority right information is stored in the priority  
16 right information field 310, thereby acquiring the contended  
17 resource 60. On the other hand, if anonymous state information is  
18 stored in the priority right information field 310, the prioritized  
19 information processing unit 340 exclusively writes information for  
20 identifying the prioritized information processing unit 340 in the  
21 priority right information field 310.

22         On the other hand, if the mode type information stored in the

1 mode type field 330 indicates the normal mode, the prioritized  
2 information processing unit 340 acquires the contended resource 60  
3 by notifying the monitor control unit 410 that the prioritized  
4 information processing unit 340 tries to acquire the contended  
5 resource 60, instead of the above-described process.

6           Moreover, if the mode type information stored in the mode type  
7 field 330 indicates the priority mode, the prioritized information  
8 processing unit 340 releases the contended resource 60 by removing  
9 resource information from the recursive acquisition phase field 320.  
10 On the other hand, if the mode type information stored in the mode  
11 type field 330 indicates the normal mode, the prioritized  
12 information processing unit 340 releases the contended resource 60  
13 by transmitting directions to release the contended resource 60 to  
14 the monitor control unit 410.

15           The non-prioritized information processing unit 350-1 has a  
16 prioritized information processing unit stop unit 360, a priority  
17 right removal unit 370, a transient state check unit 380, and an  
18 execution state setting unit 400. When the non-prioritized  
19 information processing unit 350-1 acquires the contended resource  
20 60, the prioritized information processing unit stop unit 360 first  
21 stops the prioritized information processing unit 340. Next, the  
22 prioritized information processing unit stop unit 360 notifies the

1 priority right removal unit 370 and the transient state check unit  
2 380 that the prioritized information processing unit stop unit 360  
3 has stopped the prioritized information processing unit 340.

4 For example, when the information processing units are  
5 threads, the prioritized information processing unit stop unit 360  
6 may stop the prioritized information processing unit 340 by calling  
7 the SuspendThread function. Instead of this, with an interrupt  
8 handler previously provided in the prioritized information  
9 processing unit 340, the prioritized information processing unit  
10 stop unit 360 suspends the process of the prioritized information  
11 processing unit 340 by interrupting the prioritized information  
12 processing unit 340. In this case, the processes handled by the  
13 priority right removal unit 370, the transient state check unit 380,  
14 and the execution state setting unit 400 may be executed in the  
15 interrupt handler of the prioritized information processing unit  
16 340.

17 When the priority right removal unit 370 is notified from the  
18 prioritized information processing unit stop unit 360 that the  
19 prioritized information processing unit stop unit 360 has stopped  
20 the prioritized information processing unit 340, the priority right  
21 removal unit 370 acquires an exclusive access right, which is an  
22 example of a non-prioritized exclusion right, to the resource

1 information storage area 300. The priority right removal unit 370  
2 then removes priority right information from the priority right  
3 information field 310, stores information used by the monitor  
4 control unit 410 in the priority right information field 310 and the  
5 recursive acquisition phase field 320, and stores mode type  
6 information indicating the normal mode in the mode type field 330.

7 When the transient state check unit 380 is notified from the  
8 prioritized information processing unit stop unit 360 that the  
9 prioritized information processing unit stop unit 360 has stopped  
10 the prioritized information processing unit 340, the transient state  
11 check unit 380 transmits directions to acquire the execution state  
12 of the prioritized information processing unit 340 to the execution  
13 state acquisition/notification unit 390. If the transient state  
14 check unit 380 has checked, based on the execution state received  
15 from the execution state acquisition/notification unit 390, that the  
16 stopped prioritized information processing unit 340 is executing the  
17 second process, i.e. that the stopped prioritized information  
18 processing unit 340 is executing a process for acquiring or  
19 releasing the contended resource 60 without using an indivisible  
20 instruction, such as a compare-and-swap instruction, then the  
21 transient state check unit 380 transmits the check result to the  
22 execution state setting unit 400.



1           When the execution state acquisition/notification unit 390  
2 receives directions to acquire the execution state of the  
3 prioritized information processing unit 340, from the transient  
4 state check unit 380, the execution state acquisition/notification  
5 unit 390 acquires, from the prioritized information processing unit  
6 340, the execution state thereof, e.g. execution location  
7 information indicating the execution location of a program in the  
8 prioritized information processing unit 340. The execution state  
9 acquisition/notification unit 390 then transmits the execution state  
10 of the prioritized information processing unit 340 to the transient  
11 state check unit 380, thereby notifying the non-prioritized  
12 information processing unit 350 of the execution state of the  
13 prioritized information processing unit 340. The execution location  
14 information is, for example, a program counter in the prioritized  
15 information processing unit 340. Instead of this, the execution  
16 state acquisition/notification unit 390 may acquire information for  
17 identifying a function currently being executed by the prioritized  
18 information processing unit 340, or may acquire the value of a  
19 predetermined register in the prioritized information processing  
20 unit 340.

21           When the execution state setting unit 400 has received from  
22 the transient state check unit 380 the check result indicating that

1 the prioritized information processing unit 340 is executing the  
2 second process, the execution state setting unit 400 sets the  
3 execution state of the prioritized information processing unit 340  
4 to the state where the prioritized information processing unit 340  
5 is not acquiring the contended resource 60 by using the second  
6 process. For example, the execution state setting unit 400 sets the  
7 execution state of the prioritized information processing unit 340  
8 to the state before the prioritized information processing unit 340  
9 reads the contents of the priority right information field 310. The  
10 execution state setting unit 400 then allows the prioritized  
11 information processing unit 340 to resume operations.

12 If the processes handled by the priority right removal unit  
13 370 and the execution state setting unit 400 are completed, the  
14 non-prioritized information processing unit 350-1 tries to acquire  
15 the contended resource 60 by notifying the monitor control unit 410  
16 that the non-prioritized information processing unit 350-1 tries to  
17 acquire the contended resource 60. On the other hand, the  
18 non-prioritized information processing unit 350-1 releases the  
19 contended resource 60 by notifying the monitor control unit 410 of  
20 directions to release the contended resource 60. Note that  
21 operations of the non-prioritized information processing units 350-2  
22 to 350-N are almost the same as those of the non-prioritized

1 information processing unit 350-1 and therefore a description  
2 thereof will be omitted.

3 When the monitor control unit 410 receives, from each of the  
4 prioritized information processing unit 340 and the non-prioritized  
5 information processing units 350-1 to 350-N, a notice indicating  
6 that the information processing unit tries to acquire the contended  
7 resource 60, the monitor control unit 410 allows any one of the  
8 prioritized information processing unit 340 and the non-prioritized  
9 information processing units 350-1 to 350-N to acquire the contended  
10 resource 60. In this case, the monitor control unit 410 uses the  
11 priority right information field 310 and the recursive acquisition  
12 phase field 320 as needed. When the monitor control unit 410  
13 receives directions to release the contended resource 60, from each  
14 of the prioritized information processing unit 340 and the  
15 non-prioritized information processing units 350-1 to 350-N, the  
16 monitor control unit 410 releases the contended resource 60, and  
17 allows another information processing unit waiting in order to  
18 acquire the contended resource 60, to acquire the contended resource  
19 60. An example of a process in which the monitor control unit 410  
20 arbitrates the acquisition of the contended resource 60 is almost  
21 the same as that of the monitor control unit 190 in Figure 1.  
22 Therefore, a description thereof will be omitted.

1           As described above, when only the prioritized information  
2 processing unit 340 tries to acquire the contended resource 60, the  
3 exclusion controller 20 can allow the prioritized information  
4 processing unit 340 to acquire the contended resource 60 by reading  
5 the resource information storage area 300 and writing in the  
6 resource information storage area 300 without using an indivisible  
7 instruction, such as a compare-and-swap instruction, which requires  
8 a long processing time. Therefore, the exclusion controller 20 can  
9 allow the prioritized information processing unit 340 to acquire and  
10 release the contended resource 60 at a high speed.

11           Moreover, when the non-prioritized information processing unit  
12 350 tries to acquire the contended resource 60, the exclusion  
13 controller 20 can arbitrate the acquisition of the contended  
14 resource 60 in order to ensure exclusive access to the contended  
15 resource 60.

16           Figs. 11A to 11D shows details of the resource information  
17 storage area 300 in the second embodiment. Figure 11A shows the  
18 initial state of the resource information storage area 300 when the  
19 contended resource 60 has been initialized, for example. The  
20 priority right information field 310 stores "0" indicating that  
21 priority right information is not set, i.e. indicating anonymous  
22 state information. The recursive acquisition phase field 320 stores

1 "0" indicating that the contended resource 60 is not acquired. The  
2 mode type field 330 stores "1" indicating the priority mode.

3 Figure 11C shows the state where the prioritized information  
4 processing unit 340 has acquired the contended resource 60 in Figure  
5 11A. The prioritized information processing unit 340 writes "A,"  
6 which is information for identifying the prioritized information  
7 processing unit 340, in the priority right information field 310.  
8 The prioritized information processing unit 340 writes "1"  
9 indicating that the prioritized information processing unit 340 has  
10 acquired the contended resource 60, in the recursive acquisition  
11 phase field 320. Note that the prioritized information processing  
12 unit 340 may further recursively acquire the contended resource 60  
13 having already been acquired, as shown in Figs. 11C' to 11C''.

14 Figure 11B shows the state where the prioritized information  
15 processing unit 340 has released the contended resource 60 in Figure  
16 11C. The prioritized information processing unit 340 releases the  
17 contended resource 60 by writing "0" indicating that the contended  
18 resource 60 is not acquired, in the recursive acquisition phase  
19 field 320. Note that, if it is previously predicted that the  
20 contended resource 60 is frequently acquired by a specific  
21 information processing unit, the state shown in Figure 11B may be  
22 set as the initial state of the resource information storage area

1 300. Moreover, if it is previously predicted that contention tends  
2 to occur in the acquisition of the contended resource 60, the state  
3 shown in Figure 11D may be set as the initial state of the resource  
4 information storage area 300.

5 As described above, in a reservation state, which is a state  
6 where only the prioritized information processing unit 340 tries to  
7 acquire the contended resource 60, the prioritized information  
8 processing unit 340 can execute a process for acquiring and  
9 releasing the contended resource 60 by rewriting the value in the  
10 recursive acquisition phase field 320. In this case, it is ensured  
11 that the prioritized information processing unit 340 can access the  
12 resource information storage area 300 to the exclusion of the other  
13 information processing units. Accordingly, the prioritized  
14 information processing unit 340 can acquire the contended resource  
15 60, without using a compare-and-swap instruction or the like  
16 requiring a long processing time, by read and write instructions  
17 requiring shorter processing times than the compare-and-swap  
18 instruction and the like.

19 Figure 11D shows the state where the non-prioritized  
20 information processing unit 350-1 has tried to acquire the contended  
21 resource 60 in Figure 11B or 11C. When the non-prioritized  
22 information processing unit 350-1 tries to acquire the contended

1 resource 60, the non-prioritized information processing unit 350-1  
2 acquires an exclusive access right to the resource information  
3 storage area 300. This is to avoid duplicate processes for  
4 transitioning to the normal mode by the non-prioritized information  
5 processing unit 350-1 and the other non-prioritized information  
6 processing units which may try to acquire the contended resource 60.

7         The priority right removal unit 370 stores mode type  
8 information indicating the normal mode, i.e. "0" in the mode type  
9 field 330, and also stores monitor information used by the monitor  
10 control unit 410 in the normal mode, in the priority right  
11 information field 310 and the recursive acquisition phase field 320.  
12 Instead of this, if the size of the information used in the normal  
13 mode is larger than the total size of the priority right information  
14 field 310 and the recursive acquisition phase field 320, the  
15 priority right removal unit 370 may generate a monitor structure for  
16 storing the monitor information used in the normal mode, in a  
17 predetermined area in a memory, thus storing information indicating  
18 the location of the monitor structure in the priority right  
19 information field 310 and the recursive acquisition phase field 320.

20         Figure 12 is an operation flow where the prioritized  
21 information processing unit 340 or the non-prioritized information  
22 processing unit 350 tries to acquire the contended resource in the

1 second embodiment. In some parts of this drawing, both the  
2 prioritized information processing unit 340 and the non-prioritized  
3 information processing unit 350 operate in almost the same manner.  
4 Therefore, in such parts, the prioritized information processing  
5 unit 340 and the non-prioritized information processing unit 350 are  
6 generically referred to as information processing units. When an  
7 information processing unit receives directions to acquire the  
8 contended resource 60, the information processing unit first reads  
9 the resource information storage area 300 (S800). If the  
10 information processing unit has checked, based on the read  
11 information, that the acquisition mode of the contended resource 60  
12 is set to the normal mode (S810: NO), the information processing  
13 unit tries to acquire the contended resource 60 by the monitor mode  
14 using the monitor control unit 410 (S820).

15 On the other hand, if the information processing unit has  
16 checked that the acquisition mode of the contended resource 60 is  
17 set to the priority mode (S810: YES), the information processing  
18 unit checks whether the priority right information field 310 stores  
19 anonymous state information (S830). If anonymous state information  
20 is stored therein (S830: YES), the information processing unit tries  
21 to write, in the resource information storage area 300, priority  
22 right information indicating that the information processing unit is



1 the prioritized information processing unit 340 and resource  
2 information indicating that the information processing unit has  
3 acquired the contended resource 60, to the exclusion of the other  
4 information processing units (S840). In other words, the exclusion  
5 controller 20 sets as the prioritized information processing unit  
6 340 the information processing unit which has acquired the contended  
7 resource 60 first, and sets as the non-prioritized information  
8 processing units 350 the other information processing units except  
9 the prioritized information processing unit 340. If the priority  
10 right information and the resource information have been  
11 successfully written (S850: YES), the information processing unit  
12 acquires the contended resource 60 (S860) to execute other process  
13 using the contended resource 60. If the information processing unit  
14 has failed to write the priority right information and the resource  
15 information (S850: NO), the information processing unit returns the  
16 process to S800.

17 If the priority right information field 310 does not store  
18 anonymous state information (S830: NO), the information processing  
19 unit checks whether the information processing unit is the  
20 prioritized information processing unit 340, that is, whether  
21 information for identifying the information processing unit is  
22 stored in the priority right information field 310 (S870). If the

1 information processing unit is not the prioritized information  
2 processing unit 340 (S870: NO), the non-prioritized information  
3 processing unit 350 executes a process for removing the priority  
4 right of the prioritized information processing unit 340 (S880), and  
5 then tries to acquire the contended resource 60 by the monitor mode  
6 using the monitor control unit 410 (S890). On the other hand, if  
7 the information processing unit is the prioritized information  
8 processing unit 340 (S870: YES), the prioritized information  
9 processing unit 340 checks whether the recursive acquisition phase  
10 of the contended resource 60 has reached a limit (S900). If the  
11 recursive acquisition phase has reached the limit (S900: YES), the  
12 prioritized information processing unit 340 transitions to the  
13 aforementioned process of S880. In other words, if the recursive  
14 acquisition phase of the contended resource 60, which the  
15 prioritized information processing unit 340 is trying to newly  
16 acquire, exceeds the limit of the recursive acquisition (e.g. the  
17 state of Figure 11C'') capable of being counted by using the  
18 recursive acquisition phase field 320 having a predetermined data  
19 size, the prioritized information processing unit 340 tries to  
20 acquire the contended resource 60 by the monitor mode tolerant of  
21 more phases.

22 If the recursive acquisition has not reached the limit (S900:

1 YES), the prioritized information processing unit 340 executes a  
2 process for updating the resource information in the recursive  
3 acquisition phase field 320, e.g. a process for adding one to the  
4 value already stored therein. Then, the prioritized information  
5 processing unit 340 then acquires the contended resource 60 (S860)  
6 to execute other processes using the contended resource 60.

7 Figure 13 shows an operation flow showing details of S880 in  
8 Figure 12. The information processing unit currently executing this  
9 flow checks whether the information processing unit is the  
10 prioritized information processing unit 340 (S920). If the  
11 information processing unit is not the prioritized information  
12 processing unit 340 (S920: NO), the information processing unit  
13 stops operations of the prioritized information processing unit 340  
14 (S930). Subsequently, the information processing unit reads the  
15 contents of the resource information storage area 300 (S940). The  
16 information processing unit checks whether the process for removing  
17 the priority right has already been completed, based on, for  
18 example, whether the mode type information stored in the mode type  
19 field 330 indicates the normal mode (S950).

20 If the process for removing the priority right has already  
21 been completed (S950: YES), the information processing unit allows  
22 the prioritized information processing unit to resume operations

1 (S955), and terminates the priority right removal process. On the  
2 other hand, if the process for removing the priority right has not  
3 been completed yet (S950: NO), the information processing unit  
4 acquires a right to access the resource information storage area 300  
5 to the exclusion of the other information processing units, and  
6 writes data necessary for a process for acquiring the contended  
7 resource 60 by the monitor mode, in the priority right information  
8 field 310 and the recursive acquisition phase field 320 (S960). If  
9 the writing has failed (S970: NO), the information processing unit  
10 returns to S940. If the writing has succeeded (S970: YES), the  
11 information processing unit checks whether the prioritized  
12 information processing unit 340 is executing the second process  
13 (S980).

14 If the prioritized information processing unit 340 is  
15 executing the second process (S980: YES), the execution state  
16 setting unit 400 sets the execution state of the prioritized  
17 information processing unit 340 to the state where the prioritized  
18 information processing unit 340 is not acquiring the contended  
19 resource 60 by using the second process (S990). For example, if the  
20 prioritized information processing unit 340 is executing a process  
21 which starts with reading resource information at S800 of Figure 12,  
22 is followed by sequentially executing S830, S870 and S900, and ends

1 with writing resource information at S910, the execution state  
2 setting unit 400 sets the execution state of the prioritized  
3 information processing unit 340 to the state before S800 is  
4 executed. Similarly, if the prioritized information processing unit  
5 340 is executing a process which starts with reading the resource  
6 information storage area 300 at S997 of Figure 14 to be described  
7 later and ends with completing the release of the contended resource  
8 at S1010, the execution state setting unit 400 sets the execution  
9 state of the prioritized information processing unit 340 to the  
10 state before S997 is executed.

11 Figure 14 shows an operation flow where the prioritized  
12 information processing unit 340 or the non-prioritized information  
13 processing unit 350 releases the contended resource in the second  
14 embodiment. In this drawing, since the respective operations of the  
15 prioritized information processing unit 340 and the non-prioritized  
16 information processing unit 350 will be collectively described, the  
17 prioritized information processing unit 340 and the non-prioritized  
18 information processing unit 350 are generically referred to as  
19 information processing units. When an information processing unit  
20 receives directions to release the contended resource 60, the  
21 information processing unit first reads the contents of the resource  
22 information storage area 300 (S997), and refers to the mode type

1 field 330, thereby deciding whether the acquisition mode of the  
2 contended resource 60 is set to the normal mode (S1000).

3 If the acquisition mode is set to the priority mode (S1000:  
4 YES), the information processing unit updates resource information  
5 by a process for subtracting one from the value in the recursive  
6 acquisition phase field 320 (S1005) to release the contended  
7 resource 60 (S1010). On the other hand, if the acquisition mode is  
8 set to the normal mode (S1000: NO), the information processing unit  
9 releases the contended resource 60 by the monitor mode using the  
10 monitor control unit 410 (S1020).

11 (Modified Example)

12 Figure 15 shows an operation flow showing details of S880 in  
13 an modified example of the second embodiment. The operation flow  
14 described in this drawing has a configuration including S1030  
15 instead of S980 in the operation flow described in Figure 13.

16 The information processing unit writes data necessary for a  
17 process for acquiring the contended resource 60 by the monitor mode,  
18 in the priority right information field 310 and the recursive  
19 acquisition phase field 320 (S960). If the writing has succeeded  
20 (S970: YES), the information processing unit invalidates a resource  
21 acquisition instruction issued by the prioritized information  
22 processing unit 340 (S1030). The resource acquisition instruction

1 means, for example, the operations of S910 described in Figure 12  
2 and S1010, i.e. a process for writing resource information in the  
3 resource information storage area in order to acquire the contended  
4 resource 60 by the prioritized information processing unit 340.  
5 Even when the prioritized information processing unit 340 is  
6 executing the second process, the information processing unit can  
7 transition to the monitor mode by invalidating the resource  
8 acquisition instruction without allowing the prioritized information  
9 processing unit 340 to acquire the contended resource 60.

10 Incidentally, if the information processing unit simply  
11 invalidates a resource acquisition instruction (S910) and a resource  
12 release instruction (S1010), the prioritized information processing  
13 unit 340 cannot correctly check whether the prioritized information  
14 processing unit 340 has acquired the contended resource 60 at the  
15 time when the process shown in Figure 12 has been completed, and  
16 whether the prioritized information processing unit 340 has released  
17 the contended resource 60 at the time when the process shown in  
18 Figure 14 has been completed. Therefore, in addition to  
19 invalidating the resource acquisition instruction, the information  
20 processing unit needs to allow the prioritized information  
21 processing unit 340 to retry the acquisition or the release of the  
22 contended resource 60.

1           Accordingly, the exclusion controller 20 realizes the  
2   invalidation of the instruction and the retrial of the acquisition  
3   of the contended resource 60 by using a conditional instruction  
4   execution function utilizing a predicate register. The conditional  
5   instruction execution function utilizing a predicate register is a  
6   function of executing an instruction only when the value of the  
7   predicate register is one. Here, the predicate register is provided  
8   so as to correspond to each instruction of a processor. In the  
9   present modified example, a predicate register 1 is previously made  
10   to correspond to the instructions to execute the processes of S910  
11   and S1010. Predicate register 1 is previously set to one. Further,  
12   an instruction to return the process to S800 is previously provided  
13   at the location executed when S910 has been invalidated. Similarly,  
14   an instruction to return the process to S997 is previously provided  
15   at the location executed when S1010 has been invalidated. The  
16   information processing unit can return the process to S800 and S997  
17   in addition to invalidating the resource acquisition instruction and  
18   the resource release instruction by setting predicate register 1 of  
19   the prioritized information processing unit 340 to zero.

20           Figure 16 shows an example of the hardware configuration of  
21   the exclusion controller 10. The exclusion controller 10 according  
22   to the present embodiment includes a CPU peripheral unit, an



1 input/output unit and a legacy input/output unit. The CPU  
2 peripheral unit includes a CPU 1000, RAM 1020, a graphic controller  
3 1075 and a display device 1080, which are mutually connected by a  
4 host controller 1082. The input/output unit includes a  
5 communication interface 1030, a hard disk drive 1040 and a CD-ROM  
6 drive 1060, which are connected to the host controller 1082 by an  
7 input/output controller 1084. The legacy input/output unit includes  
8 ROM 1010, a flexible disk drive 1050 and an input/output chip 1070,  
9 which are connected to the input/output controller 1084.

10 The host controller 1082 connects the RAM 1020 with the CPU  
11 1000 and the graphic controller 1075, which access the RAM 1020 at  
12 high transfer rates. The CPU 1000 operates based on programs stored  
13 in the ROM 1010 and the RAM 1020, and controls each unit. The  
14 graphic controller 1075 acquires image data generated by the CPU  
15 1000 or the like on a frame buffer provided in the RAM 1020, and  
16 displays the image data on the display device 1080. Instead of  
17 this, the graphic controller 1075 may include a frame buffer for  
18 storing image data generated by the CPU 1000 or the like, inside the  
19 graphic controller 1075.

20 The input/output controller 1084 connects the host controller  
21 1082, the communication interface 1030, the hard disk drive 1040,  
22 the CD-ROM drive 1060 and a storage device interface 1085, which are

1 relatively high-speed input/output devices. The communication  
2 interface 1030 communicates with other devices through a network.  
3 The hard disk drive 1040 stores programs and data used by the  
4 exclusion controller 10. The CD-ROM drive 1060 reads a program or  
5 data from a CD-ROM 1095 to provide the program or the data to the  
6 RAM 1020 through the input/output controller 1084.

7       Moreover, the ROM 1010 and relatively low-speed input/output  
8 devices, such as the flexible disk drive 1050 and the input/output  
9 chip 1070, are connected to the input/output controller 1084. The  
10 ROM 1010 stores a boot program executed by the CPU 1000 when the  
11 exclusion controller 10 is started up, programs depending on the  
12 hardware of the exclusion controller 10, and the like. The flexible  
13 disk drive 1050 reads a program or data from a flexible disk 1090 to  
14 provide the program or the data to the RAM 1020 through the  
15 input/output controller 1084. The input/output chip 1070 is  
16 connected to the flexible disk 1090 and various input/output devices  
17 through, for example, a parallel port, a serial port, a key board  
18 port, a mouse port, and the like.

19       A program provided to the exclusion controller 10 is provided  
20 by a user in the state where the program is stored on a recording  
21 medium, such as the flexible disk 1090, the CD-ROM 1095, or an IC  
22 card. The program is read from the recording medium, installed in

1 the exclusion controller 10 through the input/output controller  
2 1084, and executed in the exclusion controller 10.

3 The program installed and executed in the exclusion controller  
4 10 includes a prioritized information processing module, a  
5 non-prioritized information processing module, an acquisition check  
6 module, a prioritized information processing unit change module, a  
7 monitor control module, and a full stop module. Operations which  
8 are executed by the exclusion controller 10 actuated by each module  
9 are similar to those of the corresponding members in the exclusion  
10 controller 10 described in Figs. 1 to 9. Therefore, a description  
11 thereof will be omitted.

12 Incidentally, since the hardware configuration of the  
13 exclusion controller 20 is almost the same as that shown in Figure  
14 16, a description thereof will be omitted. A program installed and  
15 executed in the exclusion controller 20 includes a prioritized  
16 information processing module, a non-prioritized information  
17 processing module, a prioritized information processing unit stop  
18 module, a priority right removal module, a transient state check  
19 module, an execution state acquisition/notification module, an  
20 execution state setting module, and a monitor control module.  
21 Operations which are executed by the exclusion controller 20  
22 actuated by each module are similar to those of the corresponding

1 members in the exclusion controller 20 described in Figs. 10 to 15.  
2 Therefore, a description thereof will be omitted.

3       The above-described programs and modules may be stored on an  
4 external recording medium. An optical recording medium including a  
5 DVD and a PD, a magneto-optical recording medium including an MD, a  
6 tape medium, a semiconductor memory including an IC card, and the  
7 like can be used as the recording medium, in addition to the  
8 flexible disk 1090 and the CD-ROM 1095. Moreover, a storage device,  
9 such as a hard disk drive or RAM, which is provided in a server  
10 system connected to a dedicated communication network or the  
11 Internet, may be used as the recording medium to provide a program  
12 to the exclusion controller 10 through the network.

13       Although the present invention has been described by using the  
14 embodiments, the technical scope of the present invention is not  
15 limited to the scope described in the aforementioned embodiments.  
16 Various modifications and improvements can be made to the  
17 aforementioned embodiments. From the appended claims, it is  
18 apparent that aspects in which such modifications and improvements  
19 are made to the embodiments can be also included in the technical  
20 scope of the present invention.

21       According to the previously described embodiments, a storage  
22 device, a program for controlling the storage device, a method for

1 controlling the storage device, and a recording medium which are  
2 shown in the following respective items can be realized.

3 As apparent from the above description, according to the  
4 present invention, when only a specific information processing unit  
5 frequently acquires and releases a contended resource, high-speed  
6 acquisition of the contended resource is enabled while ensuring the  
7 exclusivity of acquisition of the contended resource.

8 Although the preferred embodiments of the present invention  
9 have been described in detail, it should be understood that various  
10 changes, substitutions and alternations can be made therein without  
11 departing from spirit and scope of the inventions as defined by the  
12 appended claims. These all solve the above-described problem with  
13 the previous art. This aspects can be achieved by combinations of  
14 features described in the appended independent claims. Moreover,  
15 the appended dependent claims specify more advantageous concrete  
16 examples of the present invention.

17 Variations described for the present invention can be  
18 realized in any combination desirable for each particular  
19 application. Thus particular limitations, and/or embodiment  
20 enhancements described herein, which may have particular  
21 advantages to the particular application need not be used  
22 for all applications. Also, not all limitations need be

1 implemented in methods, systems and/or apparatus including  
2 one or more concepts of the present invention.

3         The present invention can be realized in hardware, software,  
4 or a combination of hardware and software. A visualization tool  
5 according to the present invention can be realized in a centralized  
6 fashion in one computer system, or in a distributed fashion where  
7 different elements are spread across several interconnected computer  
8 systems. Any kind of computer system - or other apparatus adapted  
9 for carrying out the methods and/or functions described herein - is  
10 suitable. A typical combination of hardware and software could be a  
11 general purpose computer system with a computer program that, when  
12 being loaded and executed, controls the computer system such that it  
13 carries out the methods described herein. The present invention can  
14 also be embedded in a computer program product, which comprises all  
15 the features enabling the implementation of the methods described  
16 herein, and which - when loaded in a computer system - is able to  
17 carry out these methods.

18         Computer program means or computer program in the present  
19 context include any expression, in any language, code or notation,  
20 of a set of instructions intended to cause a system having an  
21 information processing capability to perform a particular function  
22 either directly or after conversion to another language, code or

1 notation, and/or reproduction in a different material form.

2           Thus the invention includes an article of manufacture which  
3 comprises a computer usable medium having computer readable program  
4 code means embodied therein for causing a function described above.  
5 The computer readable program code means in the article of  
6 manufacture comprises computer readable program code means for  
7 causing a computer to effect the steps of a method of this  
8 invention. Similarly, the present invention may be implemented as a  
9 computer program product comprising a computer usable medium having  
10 computer readable program code means embodied therein for causing a  
11 a function described above. The computer readable program code  
12 means in the computer program product comprising computer readable  
13 program code means for causing a computer to effect one or more  
14 functions of this invention. Furthermore, the present invention may  
15 be implemented as a program storage device readable by machine,  
16 tangibly embodying a program of instructions executable by the  
17 machine to perform method steps for causing one or more functions of  
18 this invention.

19           It is noted that the foregoing has outlined some of the more  
20 pertinent objects and embodiments of the present invention. This  
21 invention may be used for many applications. Thus, although the  
22 description is made for particular arrangements and methods, the

1 intent and concept of the invention is suitable and applicable to  
2 other arrangements and applications. It will be clear to those  
3 skilled in the art that modifications to the disclosed embodiments  
4 can be effected without departing from the spirit and scope of the  
5 invention. The described embodiments ought to be construed to be  
6 merely illustrative of some of the more prominent features and  
7 applications of the invention. Other beneficial results can be  
8 realized by applying the disclosed invention in a different manner  
9 or modifying the invention in ways known to those familiar with the  
10 art.